

REMOTE DEBUGGING DEVICE

Publication number: JP2001051871 (A)

Publication date: 2001-02-23

Inventor(s): OHIRA HIROTAKE

Applicant(s): RICOH KK

Classification:

- international: G06F3/12; G06F11/28; G06F3/12; G06F11/28; (IPC1-7): G06F11/28; G06F3/12

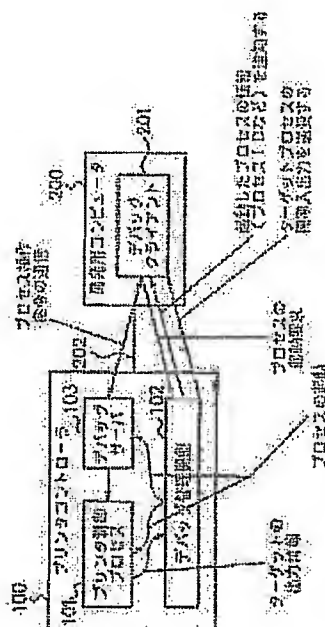
- European:

Application number: JP19990225265 19990809

Priority number(s): JP19990225265 19990809

Abstract of JP 2001051871 (A)

PROBLEM TO BE SOLVED: To perform remote debugging with simple constitution. **SOLUTION:** This remote debugging device consists of a printer controller 100 where a target program runs and a computer 200 for development which controls the printer controller 100 so as to debug the target program. The printer controller is provided with a printer control process 101 which executes the target program, a debugging server 103 which controls the printer control process at a request from the computer 200 and informs the computer 200 of the operation state of the printer control process, and a debugging management mechanism 102 which starts the debugging server and printer control process for execution under the control of the computer 200. The computer 200 is provided with a debugging client 201 which communicates with the debugging server and debugging management mechanism. Further, the debugging management mechanism informs the debugging client of standard input/output information that the target program outputs in operation.



Data supplied from the esp@cenet database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-51871

(P2001-51871A)

(43) 公開日 平成13年2月23日 (2001.2.23)

(51) Int.Cl.⁷

識別記号

F I

ターミナル* (参考)

G 0 6 F 11/28

G 0 6 F 11/28

A 5 B 0 2 1

3/12

3/12

C 5 B 0 4 2

審査請求 未請求 請求項の数 4 O L (全 11 頁)

(21) 出願番号

特願平11-225265

(22) 出願日

平成11年8月9日 (1999.8.9)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込 1 丁目 3 番 6 号

(72) 発明者 大平 浩貴

東京都大田区中馬込 1 丁目 3 番 6 号 株式

会社リコー内

Fターム (参考) 5B021 AA01 AA19 BB01 BB10

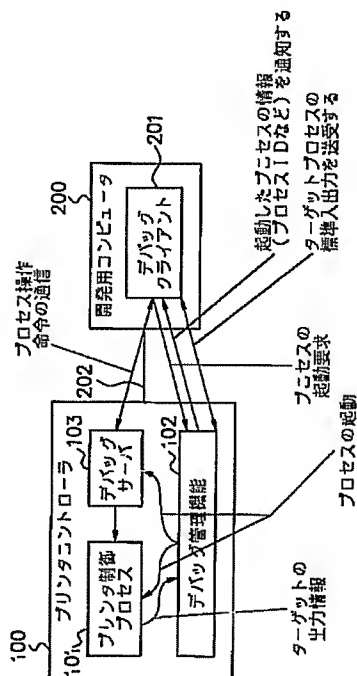
5B042 GA23 GA36 HH04

(54) 【発明の名称】 リモートデバッグ装置

(57) 【要約】

【課題】 簡単な構成によりリモートデバッグを行うことができるようにする。

【解決手段】 ターゲットプログラムが動作するプリンタコントローラ100とターゲットプログラムのデバッグを行うためにプリンタコントローラ100を制御する開発用コンピュータ200とから成る。プリンタコントローラには、ターゲットプログラムを実行するプリンタ制御プロセス101と、上記コンピュータ200からの要求に応じてプリンタ制御プロセスを制御すると共にプリンタ制御プロセスの動作状態をコンピュータ200に通知するデバッグサーバ103と、コンピュータ200の制御に応じてデバッグサーバとプリンタ制御プロセスを起動し実行させるデバッグ管理機構102とが設けられる。コンピュータ200には、上記デバッグサーバ及びデバッグ管理機構と通信するデバッグクライアント201が設けられる。また、デバッグ管理機構は、ターゲットプログラムが動作時に出力する標準入出力情報をデバッグクライアントに通知する。



【特許請求の範囲】

【請求項1】 ターゲットプログラムが動作する第1のコンピュータと、

前記ターゲットプログラムのデバッグを行うために前記第1のコンピュータを制御する第2のコンピュータと、
前記第1のコンピュータに設けられ前記ターゲットプログラムを実行するターゲットプロセスと、

前記第1のコンピュータに設けられ前記第2のコンピュータからの要求に応じて前記ターゲットプロセスを制御すると共に、前記ターゲットプロセスの動作状態を前記第2のコンピュータに通知するデバッグサーバと、

前記第1のコンピュータに設けられ前記第2のコンピュータの制御に応じて前記デバッグサーバと前記ターゲットプロセスを起動し実行させるデバッグ管理手段と、
前記第2のコンピュータに設けられ前記デバッグサーバ及び前記デバッグ管理手段と通信するデバッグクライアントとを設けたことを特徴とするリモートデバッグ装置。

【請求項2】 前記デバッグサーバは複数のターゲットプログラムを実行する複数のターゲットプロセスを制御可能とし、前記デバッグクライアントはデバッグを行うターゲットプロセスを指定すると共に、前記デバッグ管理手段は前記複数のターゲットプロセスを管理することを特徴とする請求項1記載のリモートデバッグ装置。

【請求項3】 前記デバッグ管理手段は、前記ターゲットプログラムが動作時に出力する標準出力情報を前記デバッグクライアントに通知することを特徴とする請求項1又は2記載のリモートデバッグ装置。

【請求項4】 デバッグされた新規のプログラムを送出する送出手段を前記デバッグクライアントに設けると共に、前記デバッグ管理手段は前記新規のプログラムを受け取って記憶装置に記憶させることを特徴とする請求項1又は2記載のリモートデバッグ装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ上で動作するプログラムをリモートデバッグによりデバッグするリモートデバッグ装置に関する。

【0002】

【従来の技術】コンピュータのプログラムを開発すると、プログラムに不良部分を作り込んでしまうことがある。この不良部分をバグと呼び、バグを取り除く作業をデバッグと呼ぶ。また、デバッグを支援するプログラムをデバグと呼び、デバグの対象となるプログラムをターゲットプログラムと呼ぶ。また、以下の説明において、デバグを利用してデバグする作業担当者をユーザと呼ぶ。

【0003】デバグは、ユーザの要求に応じてターゲットプログラムを動作させたり、動作状況を監視する機能を持つ。ユーザはデバグを利用してターゲットプロ

グラムを動作させ、その挙動を監視することでプログラムのバグを発見する。デバグには、さまざまな方法があるが、その一つとして例えば特開平5-20122号公報に開示されたりリモートデバグがある。

【0004】従来のリモートデバグの構成を図7に示す。リモートデバグでは、「ターゲットプロセス」と「デバグクライアント」とを二つの異なるコンピュータ上に分散して配置する。図7において、ターゲットプロセス11が存在する計算機をターゲットコンピュータ10と呼ぶ。また、デバグクライアント21が存在する計算機をホストコンピュータ20と呼ぶ。尚、30は後述する端末である。

【0005】ターゲットプロセス11とは、上述したターゲットプログラムの実行中の状態を指す。即ち、ターゲットプログラムはターゲットコンピュータ上から参照可能な記憶媒体12等の2次記憶装置上に存在し、それが主記憶装置に読み込まれて実行されている状態をターゲットプロセスと呼ぶ。尚、上記2次記憶装置とは、ターゲットコンピュータ内外を問わず、ターゲットコンピュータ上で動作すべきプログラムを固定記憶しているもので、記憶媒体12はハードディスクやROM等である。デバグはターゲットプロセス11の動作を操作し、動作状況を監視する作業となる。

【0006】また、ホストコンピュータ20におけるデバグクライアント21は、ターゲットプロセス11の実行状況をユーザに提示したり、ユーザに実行等の操作要求を入力させるプロセスであり、ユーザインターフェースの実現が主な処理である。

【0007】また、ターゲットコンピュータ10におけるデバグサーバ13は、ユーザがデバグクライアント21に入力したプロセス操作要求を通信路22を介して受け取り、その命令に応じてターゲットプロセス10を実際に実行させたり、プロセスの動作状況を取得してその情報をデバグクライアント21に送出するものである。

【0008】以上が従来提案されたりリモートデバグの構成であり、これによって、以下の利点が生じる。

・ターゲットプログラムがユーザとの入出力装置を持たない小規模なコンピュータで動作するプログラムであったとしても、通信路22さえあればホストコンピュータ20等の他のコンピュータを接続してデバグの操作を行うことができる。

【0009】・ターゲットプログラムが動作する計算機の計算能力が低い場合でも、ユーザインターフェースの処理は外部に除かれており、ターゲットコンピュータ上で必要となるデバグ処理は最小限にとどめられる。

【0010】尚、リモートデバグを含む従来のデバグの他の例のいくつかを以下に挙げておく。

・特開平5-225090号公報：遠隔デバグ方式。デバグの基本的な機能として、対象プログラムに対してブレ

クポイント(公報中では「実行停止箇所」と表記)を配置するという機能がある。このブレイクポイントの配置は従来ユーザが逐一指定していたが、これを自動化させるもの。

・特開平5-274177号公報:リアルタイムシステムデバッグ装置。デバッグのポータビリティを向上するためのデバッグの構造についての提案。

【0011】・特開平6-103104号公報:リモートデバッグ方法。この発明では、ターゲットプログラム実行の自動化のため、ターゲットプログラム上にファイルの自動実行を行うファイルを配置しておき、それをターゲット上で実行することを目的としている。尚、後述する本発明は、ファイルではなく、プログラムを起動するためのプログラム(デバッグ管理機構)を配置し、ここで、ファイルの起動だけでなく、ターゲットプログラムの入出力処理もデバッグクライアントに送出する。

【0012】特開平6-161817号公報:スレッドオンラインデバッグ装置。スレッドデバッグを対象とする。

・特開平7-306800号公報:デバッグ方式。特開平8-147189号公報:並列計算機のオペレーティングシステムのデバッグ方法。この二つの発明はカーネルデバッグの方式である。これに対して本発明はユーザプロセスのデバッグを目的としている。

【0013】・特開平8-249280号公報:データ処理装置の電子キーコンピュータに「モード」を持たせ、通常動作モードの他にサービスモード等を追加するものと思われる。サービスモードではリモートデバッグが可能となるが、この発明は「モード切替え可能な機器」の提案が主となる。

・特開平8-314760号公報:プログラム開発支援装置サーバクライアント型ソフトウェアの開発支援に関するものである。

【0014】・特開平9-120366号公報:分散アプリケーションプログラムをデバッグする分散デバッグのためのシステム及び方法。本発明が対象としている環境よりもより高機能な環境での単一のリモートデバッグシステムの構成を発明したものと思われる。本発明で提案するデバッグシステムは、従来のプログラム構成に新たな機構(デバッグ管理機構)を挿入するものであり、全体を単一の新規のシステムとして構築するわけではない。このため、本発明は上記公報の発明に比べ、従来のシステムにも適用が容易という利点がある。

【0015】・特開平9-259004号公報:デバッグ方法。ターゲットプログラム(この発明中では制御プログラム)と、デバッグサーバ(デバッグ支援プログラム)の実機への実現を前提としている。しかし、ターゲットコンピュータでは、マルチプロセスのサポートがないようなので、本発明とは対象とする環境が異なっているといえる。

【0016】・特開平9-319614号公報:リモート型デバ

ッグ方式。デバッグクライアントでのインターフェース向上に主眼を置いた発明であり、特にターゲットプロセスの動作状況表示を向上させる発明である。

・特開平10-105410号公報:インターネットイーネブル型ポートフォリオ・マネージャ・システムおよび方法。特定の環境(JAVA、HTTP等)に依存したものである。

・特表9-512404号公報:ATM交換システム。ATMスイッチに関する発明である。

【0017】

【発明が解決しようとする課題】図7のリモートデバッグのような環境を実現するためには、デバッグを行う前にターゲットコンピュータ10から参照可能な2次記憶装置上にあるデバッグサーバプログラムを主記憶装置に読み込み、プロセスとして実行しておく必要がある。しかしながら、機械制御用の小規模コンピュータ等ではこのプログラム起動命令を入力するための入出力機器を装備していない場合が多い。

【0018】このために、図7の端末30等の機器を通信路31を介して外部に接続するか、ホストコンピュータ側で端末のソフトウェアも準備しなければならない。また、デバッグサーバ13は、純粹にターゲットプロセス11の操作を行うものであり、ターゲットプロセス11からの出力は特に外部に通知されない。

【0019】即ち、図7の従来のリモートデバッグの構成では以下に示すような不具合がある。

・端末30等を準備するコストがかかる。

・ターゲットコンピュータ10に端末用の通信路31を付加するコストがかかる。

・デバッグを行う前にデバッグサーバ13を実行する作業の手間がかかる。

【0020】また、従来のリモートデバッグの構成では単一のデバッグサーバが制御できるプロセスは一つとされていた。即ち、同時に複数のターゲットプロセスをデバッグするためには、デバッグサーバとデバッグクライアントの対が複数組必要となっていた。また、デバッグサーバが複数プロセスに対応していたとしても、その標準入出力をまとめて管理し、出力することができなかった。

【0021】しかしながら、環境によっては、ターゲットコンピュータ上で複数のプロセスが協調して動作する場合がある。これらのプロセスが標準入出力に各種メッセージを送出する場合、そのメッセージがどこから来たものかを管理し、正確にユーザに通知することでデバッグの効率は格段に上がることになる。

【0022】本発明は、上記の問題を解決するために成されたもので、簡単な構成によりリモートデバッグを行うことができるようにすることを目的としている。

【0023】

【課題を解決するための手段】上記の目的を達成するた

めに、本発明によるリモートデバッグ装置は、ターゲットプログラムが動作する第1のコンピュータと、ターゲットプログラムのデバッグを行うために第1のコンピュータを制御する第2のコンピュータと、第1のコンピュータに設けられターゲットプログラムを実行するターゲットプロセスと、第1のコンピュータに設けられ第2のコンピュータからの要求に応じてターゲットプロセスを制御すると共に、ターゲットプロセスの動作状態を第2のコンピュータに通知するデバッグサーバと、第1のコンピュータに設けられ第2のコンピュータの制御に応じてデバッグサーバとターゲットプロセスを起動し実行させるデバッグ管理手段と、第2のコンピュータに設けられデバッグサーバ及びデバッグ管理手段と通信するデバッグクライアントとを設けたものである。

【0024】また、デバッグサーバは複数のターゲットプログラムを実行する複数のターゲットプロセスを制御可能とし、デバッグクライアントはデバッグを行うターゲットプロセスを指定すると共に、デバッグ管理手段は複数のターゲットプロセスを管理するようにしてよい。

【0025】また、デバッグ管理手段は、ターゲットプログラムが動作時に出力する標準入出力情報をデバッグクライアントに通知するようにしてよい。

【0026】さらに、デバッグされた新規のプログラムを送出する送出手段をデバッグクライアントに設けると共に、デバッグ管理手段は新規のプログラムを受け取って記憶装置に記憶させるようにしてよい。

【0027】

【発明の実施の形態】以下、本発明の実施の形態を図面と共に説明する。本発明の第1の実施の形態として、プリンタを制御する制御コンピュータの開発に本発明を適用する場合について説明する。図1に本実施の形態によるリモートデバッグ装置としてのリモートデバッグの構成を示す。通常、プリンタの制御はプリンタコントローラ100と呼ばれる小規模コンピュータ上で行う。プリンタコントローラ100の内部では、プリンタ制御プロセス101が動作する。本実施の形態ではこのプロセス101をデバッグすることを目的とする。ユーザは、このコントローラ100に開発用コンピュータ200を接続し、ここからデバッグの操作を行う。

【0028】即ち、プリンタコントローラ100が図7のターゲットコンピュータとなり、開発用コンピュータ200が図7のホストコンピュータとなる。デバッグ時には、従来のリモートデバッグの構成に加えて、プリンタコントローラ上に「デバッグ管理機構103」というプロセスを配置する。

【0029】このプロセスは、開発用コンピュータ200内のデバッグクライアント201と通信路202を介して相互に通信を行い、デバッグサーバプログラムを必要に応じて2次記憶装置から読み込み、デバッグサーバ103により実行させる。このため、図7の従来のリモ

ートデバッグのように、コントローラに端末を接続してデバッグサーバを実行するなどの作業は不要となる。

【0030】また、このデバッグ管理機構102は、ターゲットプログラムとしてのプリンタ制御プログラムが動作時に出力する標準入出力情報をデバッグクライアント側に送出する機能も持つ。このため、プリンタ制御プログラムが動作中に何らかの情報を標準入出力機器に提示するように作成されていると、それらの情報は全てデバッグクライアント側に提示されることになる。

【0031】また、ターゲットプログラムがX-Window Systemのように、ウィンドウを主体としたユーザインターフェースを構築できるソフトウェアであり、そのウィンドウを外部のコンピュータに送出する機能を持っているならば、ウィンドウ表示先をホストコンピュータとして設定して起動する。これによって、ホストコンピュータ（開発用コンピュータ）にターゲットプログラムのウィンドウを表示させることができる。この機能を利用して、プリンタ制御プログラムがデバッグに有効な情報を標準入出力やX-Window Systemのウィンドウに提示するように作成されていれば、それらの情報は全てデバッグクライアント側に提示されることになり、より効率的なデバッグが可能となる。

【0032】図1の構成をより詳細に示したのが図2である。図2において、ユーザが開発用コンピュータ200をプリンタコントローラ100に接続する。ユーザはデバッグクライアントで初期にデバッグするプリンタ制御プロセス101を指定しておく。この情報に基づいてデバッグクライアント201は必要に応じて自動的にデバッグサーバ103や、デバッグするプリンタ制御プロセス101の生成を要求する。この要求はデバッグ管理機構102が受け取り、ここでターゲットプロセス（プリンタ制御プロセス101）やデバッグサーバ103を実際に起動し実行させる。

【0033】また、上記ターゲットプロセスを実行する時には、ターゲットプロセスの標準入出力をデバッグサーバ103にそのまま通知するように設定しておく。これによって、標準入出力に提示される情報が全てデバッグクライアント201に通知されることとなる。

【0034】デバッグ管理機構102は、プリンタ制御プロセス101を実行したらそのプロセスを特定する情報をデバッグクライアント201に通知しておく。デバッグクライアント201はこの情報をデバッグサーバ103に通知する。以降は通常のデバッグと同様の手順をふみ、ユーザがデバッグクライアント201に対してプロセス操作要求を入力し、それをデバッグサーバ103が受け取って、プリンタ制御プロセス101を操作する。

【0035】また、プリンタ制御プログラム101の実行が完了するか、途中で強制終了された場合は、この情報は通常のリモートデバッグと同様にデバッグサーバ1

03が受け取り、デバッグクライアント201の方に通知される。この情報をデバッグクライアント201はユーザに対して提示する。ここで、新たにもう一度デバッグを繰り返すのなら、更にデバッグクライアント201がデバッグ管理機構102に対してプリンタ制御プロセス101の実行を要求する。

【0036】以上をまとめると、本実施の形態は以下のような構成、動作を有する。

- ・従来のリモートデバッグの環境に対して、新たにデバッグ管理機構102と呼ぶ部分を追加する。

- ・デバッグ管理機構102は、デバッグクライアント201の要求に応じてデバッグサーバ103の起動を行う。

- 【0037】・デバッグ管理機構102は、デバッグクライアント201の要求に応じてプリンタ制御プロセス(ターゲットプロセス)101の起動を行う。

- ・デバッグ管理機構102に対してプログラムの起動を要求する処理をデバッグクライアント201に追加する。

- ・プリンタ制御プロセス101からの標準入出力へのアクセスをデバッグクライアント201に通知する。

- 【0038】プリンタコントローラ上で動作するプリンタ制御プロセス101のデバッグに本発明を適用することにより、ターゲットプロセスやデバッグサーバといったプリンタコントローラ上でのプログラムの起動処理は全て自動で行うことができるようになる。

- 【0039】本実施の形態を適用することで以下の利点が生じる。

- ・デバッグサーバの起動を自動化することができる。

- ・プリンタ制御システムには特に端末を接続する必要がなくなる。このため、プリンタコントローラ上に、端末との通信路を配置する必要がなくなる。

- ・プリンタ制御プログラムにプログラムの動作状況を標準出力へ出力する機能を追加しておけば、この出力は全てデバッグクライアント側に自動通知される。

【0040】図3は本発明の第2の実施の形態を示す図であり、図1と対応する部分には同一番号を付して重複する説明は省略する。第1の実施の形態で説明したシステム構成では、複数のターゲットプロセスを同時にデバッグサーバ103やデバッグ管理機構102で管理することを前提としていないが、本実施の形態は、プリンタコントローラ100は、複数のプリンタ制御プロセス101を協調して動作させて、目的のプリンタ制御を実現するものである。

【0041】即ち、図3に示すように単一のデバッグサーバ103とデバッグクライアント201との間で複数のプリンタ制御プロセス101をデバッグできるようにすることを目的とする。

【0042】この場合、デバッグクライアント201とデバッグサーバ103との通信は通常のターゲットプロ

グラム操作情報だけでなく、複数あるプリンタ制御プロセス101のどれをデバッグするかを指定する情報が必要となる。デバッグサーバ103はここで指定されたプリンタ制御プロセス101を操作することになる。

【0043】また、複数のターゲットプロセスからの標準入出力を取得し、単一の通信路202を介して送受信する機構も設ける。この手順の詳細を図4に示す。本実施の形態での各種プロセスの起動は第1の実施の形態で示した手順と似ている。デバッグクライアント201は、プリンタ制御プロセス101の実行が必要かどうかを自動的に認識し、必要であればプリンタ制御プロセス101の実行をデバッグ管理機構102に要求する。

【0044】デバッグ管理機構102は、その起動したプロセスの情報(プロセスIDなど)をデバッグクライアント201に送出する。本実施の形態のデバッグクライアント201は、このようにして取得したプロセスの情報を内部に保持する。プリンタ制御プロセス101は同時に複数動作するので、保持すべきプロセスの情報は複数発生することになる。デバッグクライアント201はこれらを一括して管理する。

【0045】以降は第1の実施の形態で説明した動作と同じ動作をする。即ち、

1. ユーザがプリンタ制御プロセス101の操作要求をデバッグクライアント201に入力する。
2. デバッグクライアント201がプロセス操作要求をデバッグサーバ103に送出する。
3. デバッグサーバ103がそれに応じてプリンタ制御プロセス101の操作を実現する。

【0046】しかし、この時、デバッグ対象のプロセスは複数あり、デバッグサーバ103はどのプリンタ制御プロセス101を操作するのが不明となる。この情報は、デバッグサーバ103とデバッグ管理機構102との間に図4に示すように共有メモリ104を配置し、そこに操作すべきプロセスの情報を配置しておくことで解決する。即ち、ユーザがデバッグするプロセスを指定、あるいは変更することをデバッグクライアント201に入力すると、デバッグクライアント201はその要求をデバッグ管理機構102に通知する。

【0047】デバッグ管理機構102では、新たなデバッグ対象となるプロセスの情報を共有メモリ104に書き込む。デバッグサーバ103にプロセス操作命令が通知されると、デバッグサーバ103は共有メモリ104で示しているプロセスに対して通知されたプロセス操作命令を実現する。

【0048】以上のように、本実施の形態の構成、動作をまとめると、

- ・デバッグ管理機構102では、複数のプリンタ制御プロセスを実行できるようにする。

- ・デバッグクライアント201では、起動されたプリンタ制御プロセス101の情報を一括して管理する。

・デバッグ管理機構102とデバッグサーバ103との間に共有メモリ104を設けて、そこにどのプロセスをデバッグするかの情報を書き込む。

・デバッグサーバは共有メモリ104に書かれた情報が示しているプロセスを操作する。

・ターゲットプロセス全ての標準入出力を一括して送出する。

【0049】従って、本実施の形態を採用することで、以下のような利点が得られる。

・複数のプロセスを単一のデバッグ管理機構102とデバッグクライアント201との対でデバッグすることができる。

・あるプロセスのデバッグ中に他のプロセスをデバッグすることができる。

・デバッグしているプロセスの情報はデバッグクライアント内に一括して保存されているため、それらをまとめて処理したり、個々に取捨選択してユーザに提示することができる。

【0050】次に、本発明の第3の実施の形態を説明する。本実施の形態は、上記第1、第2の実施の形態において、ターゲットコンピュータ（プリンタコントローラ）上で動作するプログラムのデバッグを行い、バグを見付け、プログラムからそれを取り除いた後の作業を対象とする。

【0051】プログラムのバグが判明し、それを取り除いた修正済の新たなプログラムをターゲットコンピュータに送出することを考える。この新たなプログラムをターゲットコンピュータから参照可能な2次記憶装置に保存する手段として以下のものが考えられる。

1. 修正した新たなプログラムを2次記憶に書き込んでそれをターゲットコンピュータに接続する。

2. 外部のコンピュータからプログラムをターゲットコンピュータに送出する。ターゲットコンピュータはそれを受け取り、2次記憶装置上に書き込む。

【0052】1. の手段は、複雑な機構を必要とせず、実現しやすいという特徴があるが、非常に手間のかかる作業である。

2. の手段は、簡便な方法であるが、ターゲットコンピュータに対してプログラムを送出するために、外部で新たなプログラムを用意する必要がある。また、ターゲットコンピュータ上で動作するOSによっては、あるプログラムを実行中に2次記憶上のプログラムが変更されると異常な動作をするため、それを回避しなければならない。従って、本実施の形態は、上記の問題を解決するものである。

【0053】図5は本実施の形態を示すもので、図1と対応する部分には同一番号を付して重複する説明は省略する。上述したように、本実施の形態は第1、第2の実施の形態で示したデバッグ構成を利用して、デバッグを行い、バグを見付け、それをターゲットコンピュータ以

外のコンピュータで修正して新たなプリンタ制御プログラムが出来上がった状態を前提とする。

【0054】修正が完了したプログラムは他のコンピュータ（開発用コンピュータ200）上にあり、それをターゲットコンピュータ（プリンタコントローラ100）から参照可能な2次記憶装置に送出する必要がある。本実施の形態は、デバッグクライアント201とデバッグ管理機構102との間でプログラムの転送ができるようにすることを主旨とする。即ち、デバッグクライアント201にターゲットコンピュータ100に対して新規のプリンタ制御プログラムを送出する機能を付加すると共に、デバッグ管理機構102にそれを受け取り、それを2次記憶装置としての記憶媒体105に保存する機能を付加する。

【0055】即ち、

・デバッグクライアント201が参照可能な場所に、新たなプログラムを配置する。

・デバッグクライアント201がそのプログラムをデバッグ管理機構102に送出する。

・デバッグ管理機構102はそれを受け取り、記録媒体105に書き込む。

以上のようにして、完成したプログラムをターゲットコンピュータに送出する作業が自動化される。

【0056】また、もし新たなプログラムを以前からのプログラムに上書きするような場合は、図6に示すように、その以前からあるプログラムが動作中かどうかをデバッグ管理機構102が確認し、動作中であれば、書き込むのを止めてデバッグクライアント201にその旨を通知する。デバッグクライアント201はそれに応じてエラーメッセージをユーザに通知する。

【0057】以上のように、本実施の形態はターゲットプログラムを送出する機能をデバッグクライアント201とデバッグ管理機構102に付加することを提案する。これによって、以下のような利点が生じる。

・デバッグクライアント201が参照可能な位置に新規のプログラムを配置すれば、後は自動的にターゲットコンピュータ上で実行可能な状態にする。このため、新規プログラムをターゲットコンピュータ上で実行可能となるように登録する手間を省くことができる。

【0058】・書き込み時に上書きが発生する場合、プリンタコントローラ100で実行中かどうかを自動的に判断できる。

・プログラムが実行中かどうかの判断をプリンタコントローラ上で行うため、他のコンピュータで判断して他のコンピュータから書き込むよりも簡便で効率的である。

・プログラムの送信手順（通信プロトコル）は、デバッグ管理機構102とデバッグクライアント201の内部で行うものであり、そのほかのプログラムを必要とせず、またプロトコルを変更する場合もこの部分のみを変更すればよい。

【0059】

【発明の効果】以上説明したように、従来のリモートデバッグ装置では、ユーザがデバッグサーバを実行したり、直接ターゲットプロセスを指定する必要があったが、請求項1の発明によれば、ターゲットプロセスやデバッグサーバの起動を可能とするソフトウェア的機構としてのデバッグ管理手段をターゲットコンピュータ上に設けたことにより、

【0060】・デバッグすべきプリンタ制御プロセス等のターゲットプロセスや、デバッグサーバの起動を自動化することができる。

・プリンタ制御システム等のターゲットコンピュータには特に端末を接続する必要がなくなり、このため、ターゲットコンピュータ上に、端末との通信路を設ける必要がなくなる等の効果が得られる。

【0061】また、請求項2の発明によれば、一つのデバッグサーバとデバッグクライアントの対で複数のターゲットプログラムを同時にデバッグすることできると共に、複数のターゲットプロセスの情報を一元管理し、この情報を必要に応じてデバッグサーバに通知することで自在にターゲットプロセスを切替えることができるようになる。

【0062】これによって、

・複数のプロセスを単一のデバッグ管理手段、デバッグクライアントの対でデバッグすることができる。

・あるプロセスのデバッグ中に他のプロセスをデバッグすることができる。

・デバッグしているプロセスの情報はデバッグクライアント内に一括して保存されているため、それらをまとめて処理したり、ユーザに提示することができる。

・ターゲットプロセスの入出力は一括してデバッグクライアントに送出される。

【0063】また、請求項3の発明によれば、ターゲットプロセスが標準入出力に対して入出力を行うとそれらは全てデバッグクライアントに通知される。

【0064】また、請求項4の発明によれば、従来、タ

ーゲットコンピュータ外で作成したプログラムをターゲットコンピュータに送出する処理は手間がかかるものであったが、デバッグ管理手段とデバッグクライアントにプログラムの送受信機能を追加したことにより、

・プログラムをターゲットコンピュータに送出する作業が自動化できる。

・書き込みの排他制御（参照中のファイルには上書きしないようにする）が簡便に実現できる。

・プログラムの送受信機構を限られた部分に配置するために、保守が容易となる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態によるリモートデバッグ装置の構成を示すブロック図である。

【図2】本発明の第1の実施の形態によるリモートデバッグ装置の動作を示す構成図である。

【図3】本発明の第2の実施の形態によるリモートデバッグ装置の構成を示すブロック図である。

【図4】本発明の第2の実施の形態によるリモートデバッグ装置の動作を示す構成図である。

【図5】本発明の第3の実施の形態によるリモートデバッグ装置の構成を示すブロック図である。

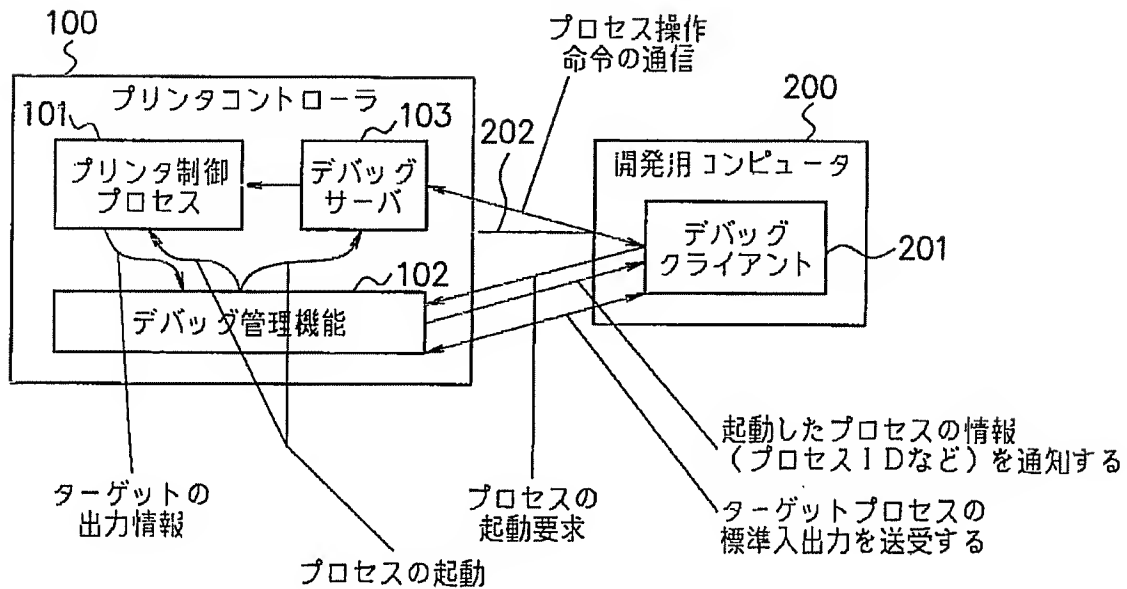
【図6】本発明の第3の実施の形態によるリモートデバッグ装置の動作を示す構成図である。

【図7】従来のリモートデバッグ装置の構成を示すブロック図である。

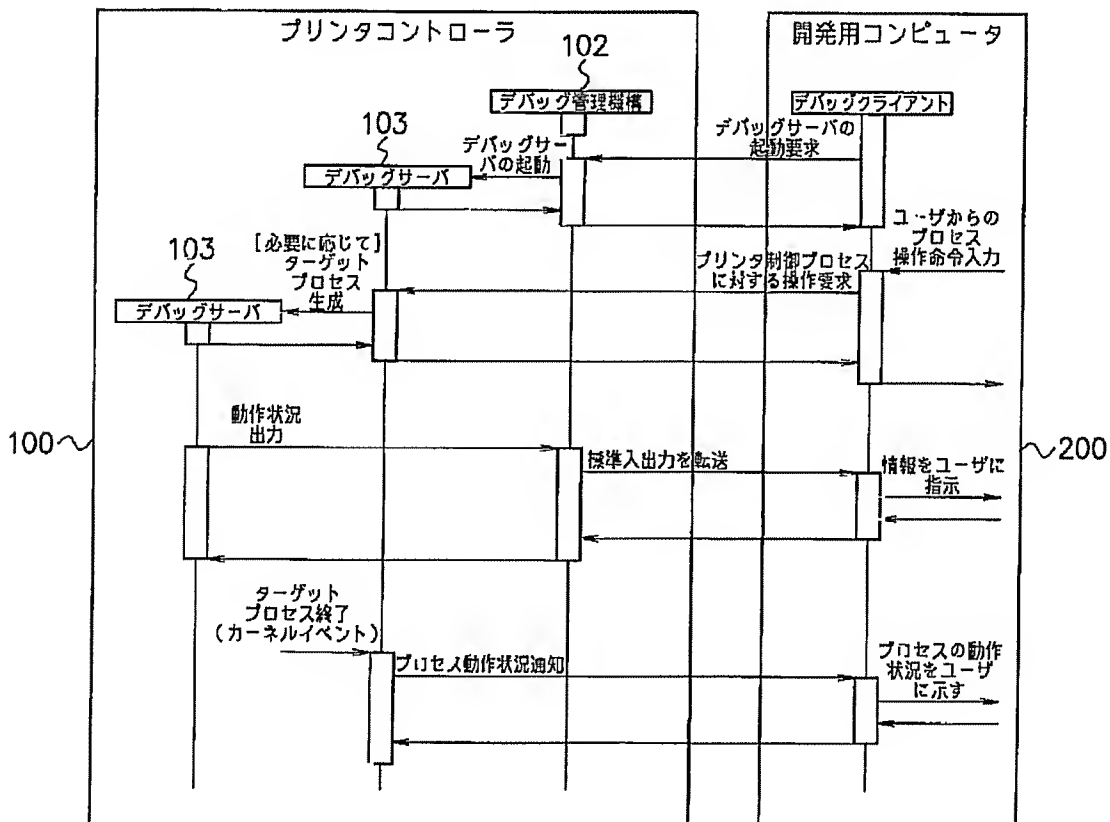
【符号の説明】

- 100 プリンタコントローラ
- 101 プリンタ制御プロセス
- 102 デバッグ管理機構
- 103 デバッグサーバ
- 104 共有メモリ
- 105 記憶媒体
- 200 開発用コンピュータ
- 201 デバッグクライアント
- 202 通信路

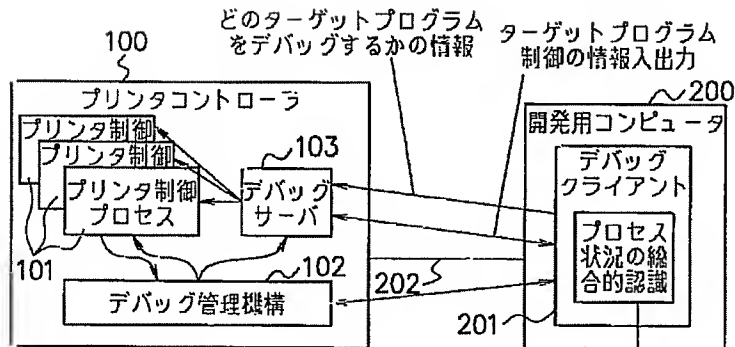
【図1】



【図2】

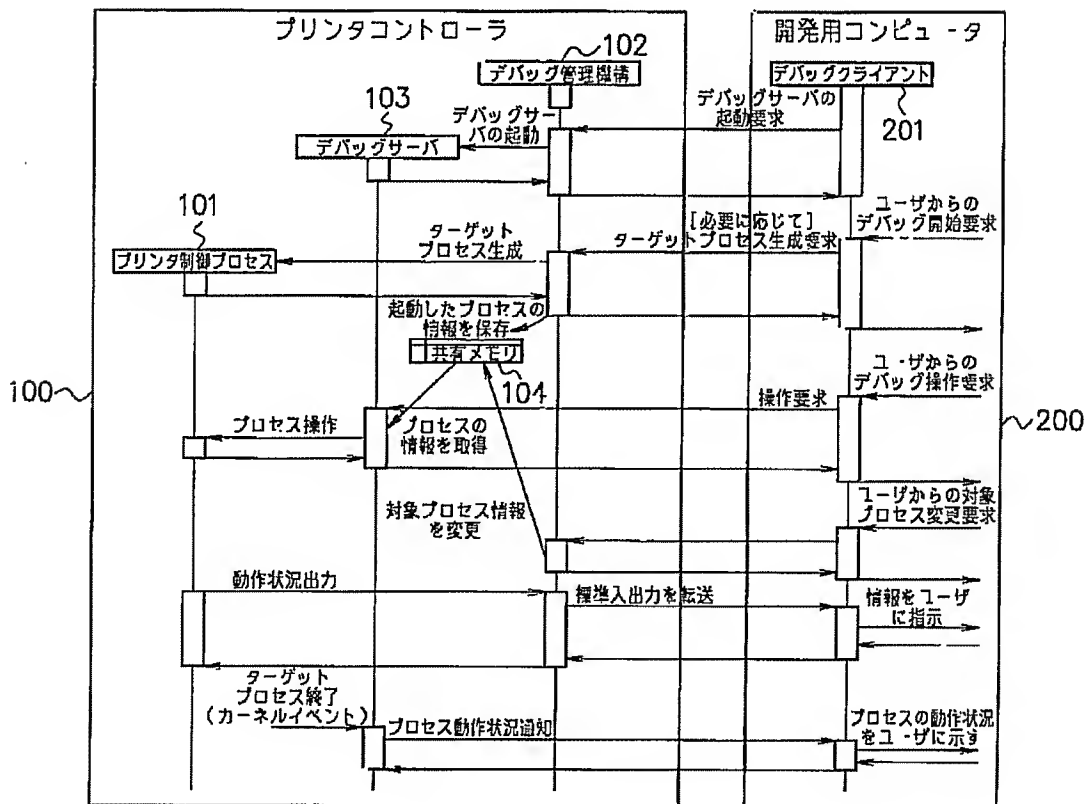


【図3】

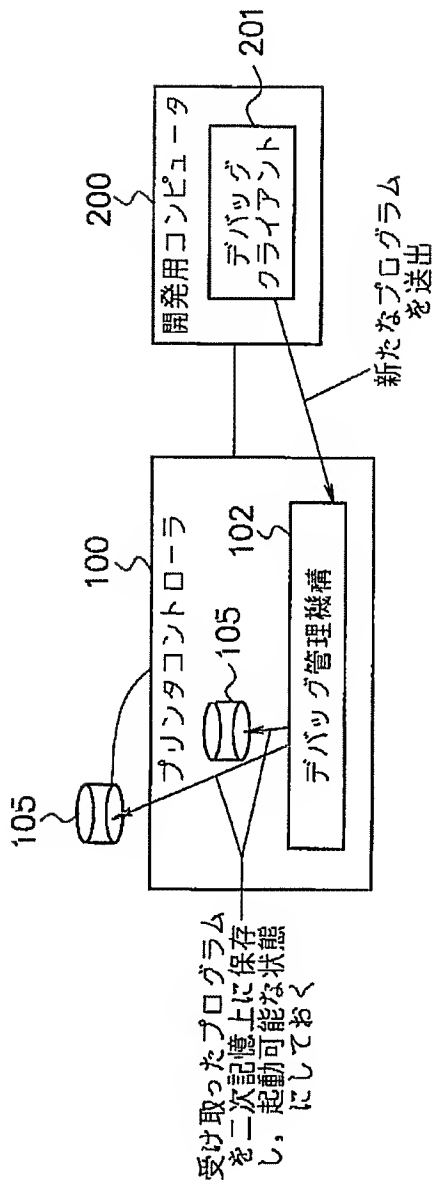


プロセスの動作状況を出力し、それに応じてユーザがデバッグ対象プログラムを動的に切替える

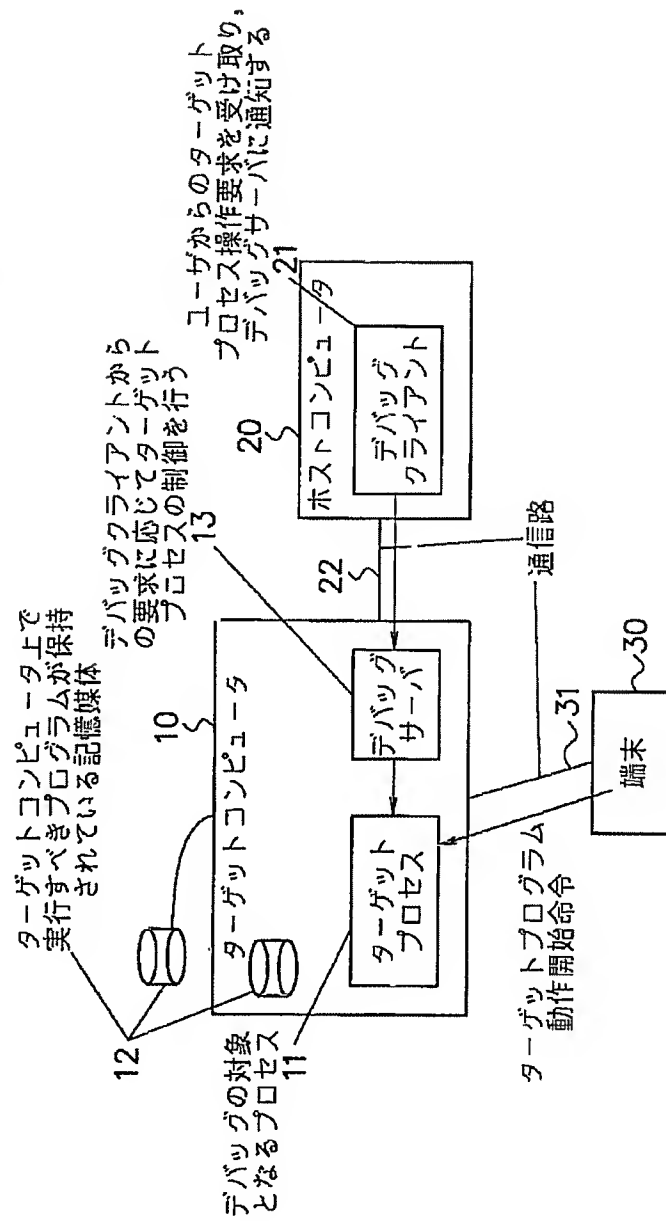
【図4】



【図5】



【図7】



【図6】

